METHODS AND APPARATUS FOR GENERATING A DATA CLASSIFICATION MODEL USING AN ADAPTIVE LEARNING ALGORITHM

Cross Reference to Related Application

The present invention is related to United States Patent Application entitled "Method and Apparatus for Generating a Data Classification Model Using Interactive Adaptive Learning Algorithms," (Attorney Docket Number YOR920000507US1), filed contemporaneously herewith, assigned to the assignee of the present invention and incorporated by reference herein.

Field of the Invention

The present invention relates generally to the fields of data mining or machine learning and, more particularly, to methods and apparatus for generating data classification models.

Background of the Invention

Data classification techniques, often referred to as supervised learning, attempt to find an approximation or hypothesis to a target concept that assigns objects (such as processes or events) into different categories or classes. Data classification can normally be divided into two phases, namely, a learning phase and a testing phase. The learning phase applies a learning algorithm to training data. The training data is typically comprised of descriptions of objects (a set of feature variables) together with the correct classification for each object (the class variable).

The goal of the learning phase is to find correlations between object descriptions to learn how to classify the objects. The training data is used to construct models in which the class variable may be predicted in a record in which the feature variables are known but the class variable is unknown. Thus, the end result of the learning phase is a model or hypothesis (e.g., a set of rules) that can be used to predict the

class of new objects. The testing phase uses the model derived in the training phase to predict the class of testing objects. The classifications made by the model is compared to the true object classes to estimate the accuracy of the model.

Numerous techniques are known for deriving the relationship between the feature variables and the class variables, including, for example, Disjunctive Normal Form (DNF) Rules, decision trees, nearest neighbor, support vector machines (SVMs) and Bayesian classifiers, as described, for example, in R. Agrawal et al., "An Interval Classifier for Database Mining Applications," Proc. of the 18th VLDB Conference, Vancouver, British Columbia, Canada 1992; C. Apte et al., "RAMP: Rules Abstraction for Modeling and Prediction," IBM Research Report RC 20271, June 1995; J.R. Quinlan, "Induction of Decision Trees," Machine Learning, Volume 1, Number 1, 1986; J. Shafer et al., "SPRINT: A Scaleable Parallel Classifier for Data Mining," Proc. of the 22d VLDB Conference, Bombay, India, 1996; M. Mehta et al., "SLIQ: A Fast Scaleable Classifier for Data Mining," Proceedings of the Fifth International Conference on Extending Database Technology, Avignon, France, March, 1996, each incorporated by reference herein.

Data classifiers have a number of applications that automate the labeling of unknown objects. For example, astronomers are interested in automated ways to classify objects within the millions of existing images mapping the universe (e.g., differentiate stars from galaxies). Learning algorithms have been trained to recognize these objects in the training phase, and used to predict new objects in astronomical images. This automated classification process obviates manual labeling of thousands of currently available astronomical images.

While such learning algorithms derive the relationship between the feature variables and the class variables, they generally produce the same output model given the same domain dataset. Generally, a learning algorithm encodes certain assumptions about the nature of the concept to learn, referred to as the bias of the learning algorithm. If the

assumptions are wrong, however, then the learning algorithm will not provide a good approximation of the target concept and the output model will exhibit low accuracy. Most research in the area of data classification has focused on producing increasingly more accurate models, which is impossible to attain on a universal basis over all possible domains. It is now well understood that increasing the quality of the output model on a certain group of domains will cause a decrease of quality on other groups of domains. See, for example, C. Schaffer, "A Conservation Law for Generalization Performance," Proc. of the Eleventh Int'l Conference on Machine Learning, 259-65, San Francisco, Morgan Kaufman (1994); and D. Wolpert, "The Lack of a Priori Distinctions Between Learning Algorithms and the Existence of a Priori Distinctions Between Learning Algorithms," Neural Computation, 8 (1996), each incorporated by reference herein.

While conventional learning algorithms produce sufficiently accurate models for many applications, they suffer from a number of limitations, which, if overcome, could greatly improve the performance of the data classification and regression systems that employ such models. Specifically, the learning algorithms of conventional data classification and regression systems are unable to adapt over time. In other words, once a model is generated by a learning algorithm, the model cannot be reconfigured based on experience. Thus, the conventional data classification and regression systems that employ such models are prone to repeating the same errors.

A need therefore exists for data classification and regression methods and apparatus that adapt a learning algorithm through experience. Another need exists for data classification and regression methods and apparatus that dynamically modify the assumptions of the learning algorithm to improve the assumptions embodied in the generated models and thereby improve the quality of the data classification and regression systems that employ such models. Yet another need exists for a learning method and apparatus that performs meta-learning to improve the assumptions or inductive bias in a model.

Summary of the Invention

Generally, a data classification method and apparatus are disclosed for labeling unknown objects. The disclosed data classification system employs a learning algorithm that adapts through experience. The present invention classifies objects in domain datasets using data classification models having a corresponding bias and evaluates the performance of the data classification. The performance values for each domain dataset and corresponding model bias are processed to initially identify (and over time modify) one or more rules of experience. The rules of experience are then subsequently used to generate a model for data classification. Each rule of experience specifies one or more characteristics for a domain dataset and a corresponding bias that should be utilized for a data classification model if the rule is satisfied.

Thus, the present invention dynamically modifies the assumptions (bias) of the learning algorithm to improve the assumptions embodied in the generated models and thereby improve the quality of the data classification and regression systems that employ such models. Furthermore, since the rules of experience change dynamically, the learning process of the present invention will not necessarily output the same model when the same domain dataset is presented again. Furthermore, the disclosed self-adaptive learning process will become increasingly more accurate as the rules of experience are accumulated over time.

A more complete understanding of the present invention, as well as further features and advantages of the present invention, will be obtained by reference to the following detailed description and drawings.

Brief Description of the Drawings

FIG. 1 is a schematic block diagram showing the architecture of an illustrative data classification system in accordance with the present invention;

FIG. 2 illustrates the operation of the data classification system;

FIG. 3 illustrates an exemplary table from the domain dataset of FIG. 1;

FIG. 4 illustrates an exemplary table from the performance dataset of FIG. 1;

FIG. 5 illustrates an exemplary table from the rules of experience table of FIG. 1;

FIG. 6 is a flow chart describing the meta-feature generation process of FIG. 1;

FIG. 7 is a flow chart describing the performance assessment process of FIG. 1;

FIG. 8 is a flow chart describing the rules of experience generation process of FIG. 1; and

FIG. 9 is a flow chart describing the self-adaptive learning process of FIG. 1 incorporating features of the present invention.

Detailed Description of Preferred Embodiments

FIG. 1 illustrates a data classification system 100 in accordance with the present invention. The data classification system 100 may be embodied as a conventional data classification system, such as the learning program described in J. R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc. Palo Alto, CA, incorporated by reference herein, as modified in accordance with the features and functions of the present invention to provide an adaptive learning algorithm.

FIG. 1 is a schematic block diagram showing the architecture of an illustrative data classification system 100 in accordance with the present invention. The data classification system 100 may be embodied as a general purpose computing system, such as the general purpose computing system shown in FIG. 1. The data classification system 100 includes a processor 110 and related memory, such as a data storage device

120, which may be distributed or local. The processor 110 may be embodied as a single processor, or a number of local or distributed processors operating in parallel. The data storage device 120 and/or a read only memory (ROM) are operable to store one or more instructions, which the processor 110 is operable to retrieve, interpret and execute. As shown in FIG. 1, the data classification system 100 optionally includes a connection to a computer network (not shown).

As shown in FIG. 1 and discussed further below in conjunction with FIGS. 3 through 5, the data storage device 120 preferably includes a domain dataset 300, a performance dataset 400 and a rules of experience table 500. Generally, the domain dataset 300 contains a record for each object and indicates the class associated with each object. The performance dataset 400 indicates the learning algorithm that produced the best model for each domain. The rules of experience table 500 identify a number of prioritized rules and their corresponding conditions, which if satisfied, provide a bias or assumption that should be employed when generating a model.

In addition, as discussed further below in conjunction with FIGS. 6 through 11, the data storage device 120 includes a meta-feature generation process 600, a performance assessment process 700, a rules of experience generation process 800 and a self-adaptive learning process 900. Generally, the meta-feature generation process 600 processes each domain dataset to represent the domain as a set of meta-features. The performance assessment process 700 evaluates the performance of a given model for a given domain dataset described by a set of meta-features and stores the results in the performance dataset 400. The rules of experience generation process 800 evaluates the performance dataset 400 in order to modify or extend the current rules in the rules of experience table 500. The self-adaptive learning process 900 identifies the best model for a given domain dataset 300, based on the current rules of experience table 500.

FIG. 2 provides a global view of the data classification system 100. As shown in FIG. 2, a domain dataset 300, discussed below in conjunction with FIG. 3,

serves as input to the system 100. The domain dataset 300 is applied to a self-adaptive learning process 900, discussed below in conjunction with FIG. 9, during step 220 and a meta-feature generation process 600, discussed below in conjunction with FIG. 6, during step 240. Generally, the self-adaptive learning process 900 produces an output model 250 that can be used to predict the class labels of future examples. For a detailed discussion of suitable models 250, see, for example, J.R. Quinlan, C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc. Palo Alto, CA. (1994) (decision trees); Weiss, Sholom and Indurkhya, Nitin, "Optimized Rule Induction", Intelligent Expert, Volume 8, Number 6, pp. 61-69, 1993 (rules); and L.R. Rivest, "Learning Decision Lists", Machine Learning, 2, 3, 229-246, (1987) (decision lists), each incorporated by reference herein.

The meta-feature generation process 600 executed during step 240 represents the domain dataset 300 as a set of meta-features. The performance of the output model 250 is assessed during step 260 by the performance assessment process 700, discussed below in conjunction with FIG. 7, and the performance assessment is recorded in the performance dataset 400. The performance assessment process 700 executed during step 260 evaluates how much the output model 250 can be improved.

As shown in FIG. 2, the self-adaptive learning process 900 receives the following information as inputs: (i) the domain dataset 300; (ii) the meta-feature description of the domain dataset 300; and (iii) the performance dataset 400. As discussed further below in conjunction with FIG. 9, the self-adaptive learning process 900 can use these inputs to modify the underlying assumptions embodied in a given model, such that, if the same dataset 300 were to be presented again to the self-adaptive learning process 900 a more accurate model would be produced.

DATABASES

FIG. 3 illustrates an exemplary table from the domain dataset 300 that includes training examples, each labeled with a specific class. As previously indicated,

the domain dataset 300 contains a record for each object and indicates the class associated with each object. The domain dataset 300 maintains a plurality of records, such as records 305 through 320, each associated with a different object. For each object, the domain dataset 300 indicates a number of features in fields 350 through 365, describing each object in the dataset. The last field 370 corresponds to the class assigned to each object. For example, if the domain dataset 300 were to correspond to astronomical images to be classified as either stars or galaxies, then each record 305-320 would correspond to a different object in the image, and each field 350-365 would correspond to a different feature such as the amount of luminosity, shape or size. The class field 370 would be populated with the label of "star" or "galaxy."

FIG. 4 illustrates an exemplary table from the performance dataset 400. As previously indicated, the performance dataset 400 indicates the performance for each model on a domain. The performance dataset 400 maintains a plurality of records, such as records 405 through 415, each associated with a different model. For each model, the performance dataset 400 identifies the domain on which the model was utilized in field 450, as well as the underlying bias embodied in the model in field 455 and the performance assessment in field 460. Each domain can be identified in field 450, for example, using a vector of meta-features characterizing each domain (as produced by the meta-feature generation process 600).

FIG. 5 illustrates an exemplary table from the rules of experience table 500. The rules of experience table 500 identifies a number of prioritized rules and their corresponding conditions, which if satisfied, provide a bias or assumption that should be employed when generating a model. As shown in FIG. 5, the rules of experience table 500 includes a plurality of records, such as records 505 through 515, each associated with a different experience rule. For each rule identified in field 550, the rules of experience table 500 identifies the corresponding conditions associated with the rule in field 560 and

the bias or assumption that should be employed in a model when the rule is satisfied in field 570.

PROCESSES

FIG. 6 is a flow chart describing the meta-feature generation process 600. As previously indicated, the meta-feature generation process 600 processes each set of domain data to represent the domain as a set of meta-features. As shown in FIG. 6, the meta-feature generation process 600 initially processes the domain dataset 300 during step 610 to store the information in a table. Thereafter, the meta-feature generation process 600 extracts statistics from the dataset 300 during step 620 that are then used to generate meta-features during step 630. For a discussion of the generation of meta-features that are particularly relevant to the meta-learning phase, including concept variation or average weighted distance meta-features, as well as additional well-known meta-features, see, for example, United States Patent Application Serial Number 09/629,086, filed July 31, 2000, entitled "Methods and Apparatus for Selecting a Data Classification Model Using Meta-Learning," assigned to the assignee of the present invention and incorporated by reference herein.

FIG. 7 is a flow chart describing the performance assessment process 700. The performance assessment process 700 evaluates the performance of a given model for a given domain dataset and stores the results in the performance dataset 400. The process 700 initially receives a model 250 during step 710 and assesses empirically the performance of the model 250. In other words, the model 250 is used to classify objects during step 710, for which the classification is already known, so that an objective measure of the model performance may be obtained. Typically, the performance assessment corresponds to the estimated accuracy of the model 250.

As shown in FIG. 7, the domain is then processed during step 715 by the meta-feature generation process 600, discussed above in conjunction with FIG. 6, to obtain a vector of meta-features characterizing the domain. Thereafter, a new entry is

created in the performance dataset 400 during step 720 using (i) the meta-feature description of the domain on which the model 250 was utilized, (ii) the underlying bias embodied in the model and (iii) the performance assessment determined during step 710.

FIG. 8 is a flow chart describing an exemplary rules of experience generation process 800 that evaluates the performance dataset 400 in order to modify or extend the current rules in the rules of experience table 500. As shown in FIG. 8, the rules of experience generation process 800 initially evaluates the performance dataset 400 during step 810 to identify correlations between various domains (described by a set of meta-features) and their corresponding best inductive bias (model).

Generally, the rules of experience generation process 800 employs a simple learning algorithm that receives a domain as input (in this case, the performance dataset 400) and produces as a result a model (in this case, the rule of experience 500). The difference lies in the nature of the domain. For a simple learning algorithm, the domain is a set of objects that belong to a real-world application, and where we wish to be able to predict the class of new objects. In the rules of experience generation process 800, each object contains the meta-features of a domain and the class of each object indicates the bias used to learn that domain. The rules of experience generation process 800 is thus a meta-learner that learns about the learning process itself. The mechanism behind it, however, is no different from a simple learning algorithm.

Based on the correlations identified during step 810, the current rules of experience are modified or extended during step 820 and recorded in the rules of experience table 500. For example, as shown in the exemplary rules of experience table 500 of FIG. 5, when models used a particular bias that partitioned the data in a specified manner, certain correlations were identified in various meta-features.

The modification or extension of the rules in the rules of experience table 500 will influence the future selection of models by the self-adaptive learning process 900, discussed below in conjunction with FIG. 9. Since the rules of experience change

dynamically, the learning process 900 of the present invention will not necessarily output the same model when the same domain dataset is presented again. Furthermore, the self-adaptive learning process 900 will become increasingly more accurate as the rules of experience table 500 grows larger.

FIG. 9 is a flow chart describing an exemplary self-adaptive learning process 900 that identifies the best model for a given domain dataset 300, based on the current rules of experience table 500. As shown in FIG. 9, the self-adaptive learning process 900 initially executes the meta-feature generation process 600, discussed above in conjunction with FIG. 6, during step 910 to provide a meta-feature description of the current domain. During step 920, the self-adaptive learning process 900 sequentially compares the meta-feature description of the current domain to each of the rules in the rules of experience table 500 until a rule is satisfied. In this manner, the first satisfied rule provides the best bias to utilize for the current domain.

If a rule is satisfied, then the corresponding bias is applied to generate the model 250 during step 930. If, however, no rule in the rules of experience table 500 is satisfied for the current domain, then a default bias is retrieved during step 940 and the default bias is applied to generate the model 250 during step 930. Thereafter, program control terminates.

It is to be understood that the embodiments and variations shown and described herein are merely illustrative of the principles of this invention and that various modifications may be implemented by those skilled in the art without departing from the scope and spirit of the invention.